Using Fuzzy Logic With Microcontrollers: An Overview

Hong Soo Ng
Applications Engineer
Intel Corporation
Article ID #0502

Introduction

Fuzzy logic is a technology based on fuzzy set theory developed by Professor Lofti Zadeh in 1965. Over the last few years, it has been finding a rapidly increasing number of applications in consumer electronics and embedded control systems. Fuzzy logic provides a new way of solving problems using a human-like reasoning process rather than an exact mathematical modeling of the system. As a result, in many cases, it reduces the product development cycle and produces systems that are more robust and have more features. In addition, most fuzzy logic applications can be easily implemented using standard microcontrollers.

Fuzzy Logic vs. Conventional Technology

Solving control problems using conventional methods or algorithms requires complete knowledge of the system. Many times, the system is nonlinear or so complex that it either requires a high-order mathematical equation to model or yields no clear mathematical model. This makes the problem very difficult or even impossible to solve. Unlike conventional technology, fuzzy logic uses a rule-based system architecture that allows the system to be described by imprecise or human-like statements and variables.

In general, using fuzzy logic has the advantages of lowering product development cost; reducing time to market; and improving product performance, reliability and robustness. Fuzzy logic is a better solution than conventional technology when systems are too complex to model accurately, when they have operational nonlinearities, or when they have uncertainties in either their inputs or definition. However, fuzzy logic is not the solution to all problems. If a system can easily be accurately modeled and optimally solved with conventional technology, there is no reason to turn to fuzzy logic.

Fuzzy Logic Basics

Fuzzy logic involves three primary processes: fuzzification, fuzzy rule inference, and defuzzification (Figure 1). During fuzzification, crisp input values are translated into a degree of membership function (a fractional value from 0 to 1) in a number of fuzzy sets, using predefined input membership functions. The fuzzy inputs are then evaluated using predetermined linguistic rules to produce fuzzy outputs. Finally, all fuzzy outputs are combined and translat-

Continued on page 4

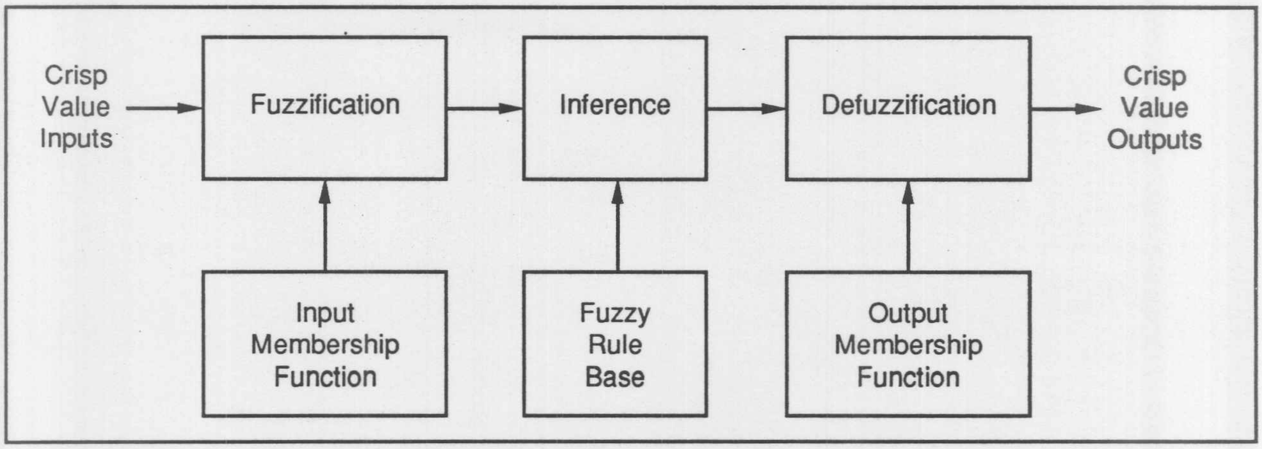

Figure 1. Fuzzy Logic Processes

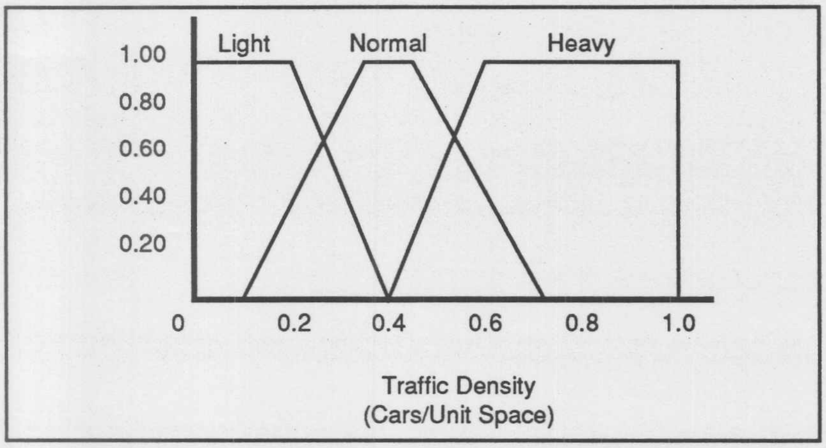

Figure 2. Input Membership Function for Traffic Density

Continued from page 3

ed into crisp and executable output values, using output membership functions during the defuzzification process.

Consider a simplified traffic control problem. The input to the system is traffic density and the output is the green light duration. Using intuition, we can easily come up with a few rules to the system:

Rule 1: If traffic is light, then green light is short.

Rule 2: If traffic is normal, then green light is medium long.

Rule 3: If traffic is heavy, then green light is long.

We notice that terms such as light, heavy, normal, short and long are all linguistic terms from everyday language, without real precision, and thus fuzzy. Unlike classical set theory, in which a crisp value can be only either a member or not a member of a given set, fuzzy logic allows a value to have degrees of membership in different sets. In this example, the input membership function for traffic density and the output membership function for green light duration are determined as in Figure 2 and Figure 3.

Assuming that a crisp input value of 0.3 comes in to the system, after fuzzification, the value will have a membership value of 0.5 to "light" and 0.8 to "normal". In other words, the traffic density is 50% light and 80% normal. This will fire Rule 1 and Rule 2 simultaneously. In the fuzzy rule inference process, the degree to which a rule's condition is true will determine the degree to which the resulting action is taken.

This is done by activating the area of output membership functions to the degree to which the rule's condition is true. In this case, it means that the action to turn on the green light for a medium long time will be executed with a greater strength than the action to turn on the green light for a short time. These two actions will be combined and then defuzzified to produce a crisp value output that can be executed. The most frequently used defuzzification method is called Center-of-Area (CoA), in which the center of area location to all applicable output membership functions is calculated to produce the crisp output value. The process is illustrated in Figure 4.

Using Fuzzy Logic with Microcontrollers

Most fuzzy logic applications are currently implemented in software using standard microcontrollers. The integrated architecture of most microcontrollers with core CPU, memory, and peripherals makes them highly desirable platforms for implementing fuzzy logic applications. Applications that are very complex and time-critical may require a dedicated fuzzy processor. However, most fuzzy logic applications work well using software implementation on standard microcontrollers. In a typical application, the fuzzy inference rule base and member-

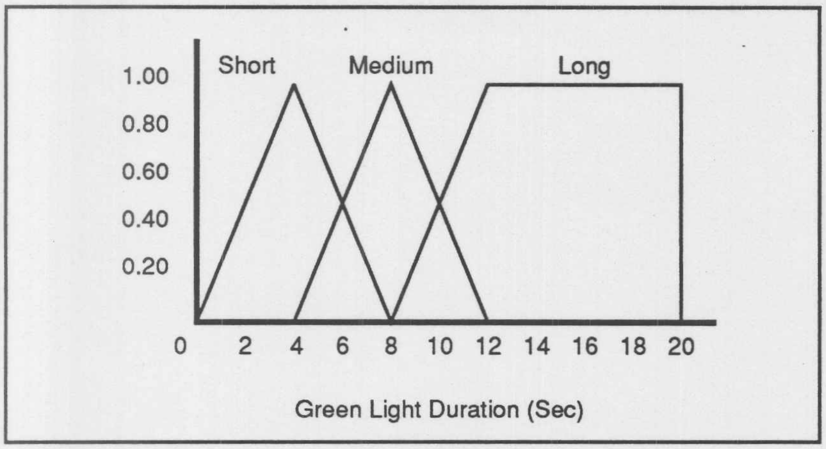

Figure 3. Output Membership Function for Green Light Duration

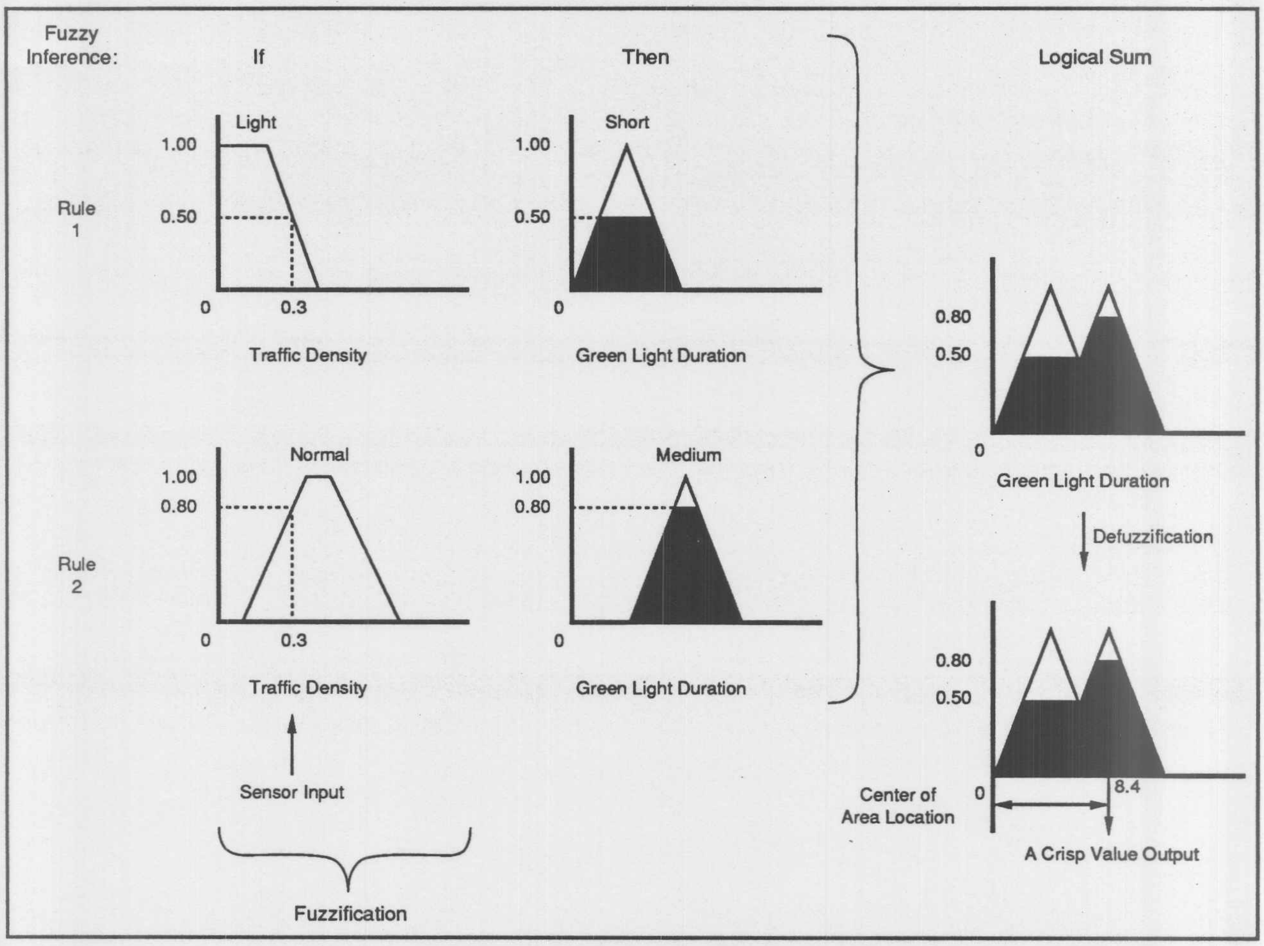

Figure 4. Fuzzy Rule Inference Process

ship function information can be stored as a look-up table in the memory available onboard most microcontrollers. Fuzzification, fuzzy rule inference, and defuzzification processes are then implemented in software using multiply and accumulate functions.

A number of fuzzy logic development tools and products are currently available in the market. These tools greatly simplify the development work by providing graphical tools for fuzzy logic design, optimization, and verification. Inform Software Corp., a pioneer of fuzzy logic applications in Europe, is currently working closely with Intel in providing fuzzy logic development tools for Intel's embedded microcontrollers and microprocessors. Inform's *fuzzy*TECH software provides a window-based, integrated graphical environment for fuzzy logic application development, debugging, simulation, and verification. It can also generate C code as well as highly optimized assembly code for Intel's MCS®-96 and MCS®-51 embedded microcontrollers. Together with this user-friendly development environment, Intel's embedded architecture provides an easy and efficient way to implement fuzzy logic applications.

**fuzzy*TECH is a trademark of Inform Software Corp.